Thin Structures in Image Based Rendering

Theo Thonat, Abdelaziz Djelouah, Fredo Durand, George Drettakis

Eurographics Symposium on Rendering 2018
T. Hachisuka and W. Jakob
(Guest Editors)

Volume 37 (2018), Number 4

Thin Structures in Image Based Rendering

Theo Thonat[1] and Abdelaziz Djelouah[1,⋆] and Fredo Durand[2,1] and George Drettakis[1]

[1]Inria, Université Côte d'Azur
[2]MIT CSAIL

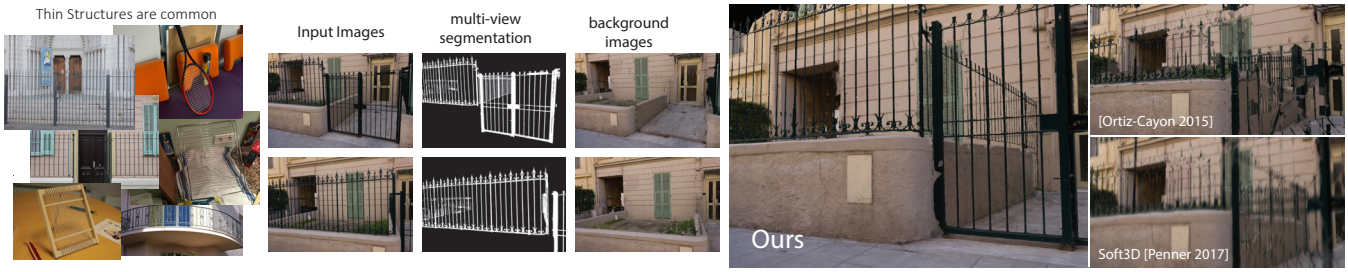

Figure 1: *Thin structures are present in many environments, both indoors and outdoors (far left). They are challenging for image-based rendering (IBR) methods since they are hard to reconstruct. We address this problem by introducing a multi-view segmentation algorithm for thin structures supported by user provided simple geometries, which exploits multi-view information. Our algorithm can handle* multiple layers *of thin structures, such as the different level of fences in the scene above. Our solution extracts multi-view mattes together with clean background images and geometry. These elements are used by our multi-layer rendering algorithm that allows* free-viewpoint *navigation, with significantly improved quality compared to previous solutions (right).*

Abstract

We propose a novel method to handle thin structures in Image-Based Rendering (IBR), and specifically structures supported by simple geometric shapes such as planes, cylinders, etc. These structures, e.g. railings, fences, oven grills etc, are present in many man-made environments and are extremely challenging for multi-view 3D reconstruction, representing a major limitation of existing IBR methods. Our key insight is to exploit multi-view information. After a handful of user clicks to specify the supporting geometry, we compute multi-view and multi-layer alpha mattes to extract the thin structures. We use two multi-view terms in a graph-cut segmentation, the first based on multi-view foreground color prediction and the second ensuring multi-view consistency of labels. Occlusion of the background can challenge reprojection error calculation and we use multiview median images and variance, with multiple layers of thin structures. Our end-to-end solution uses the multi-layer segmentation to create per-view mattes and the median colors and variance to create a clean background. We introduce a new multi-pass IBR algorithm based on depth-peeling to allow free-viewpoint navigation of multi-layer semi-transparent thin structures. Our results show significant improvement in rendering quality for thin structures compared to previous image-based rendering solutions.

1. Introduction

Image-based rendering (IBR) algorithms, e.g. [KLS*13, LKM14, OCDD15, PZ17] provide a compelling solution to virtual visits and photo tourism, avoiding the expense of 3D modeling/texturing, and complex photo-realistic rendering. Their key advantage is the simplicity of the input: only a set of photos of a scene is needed, yet they allow high-quality free-viewpoint rendering. However, one of the key problems in IBR methods is the rendering of regions where 3D reconstruction is hard or impossible, such as vegetation, reflections and *thin structures*.

Previous methods have addressed reflections and transparency [SKG*12, KLS*13], or vegetation [CDSHD13] but no solution currently exists for thin structures. These are present in all man-made environments: outdoors, fences or stair banisters are very common, while indoors a variety of everyday objects and utensils have similar properties, e.g., grills, racks or decorative elements. In this paper we focus on thin structures that are supported by user provided simple geometries, such as planes or cylinders.

We take as input a set of photographs of the scene from multiple

⋆Abdelaziz Djelouah is now at Disney Research. He contributed to this work during his time at Inria.

viewpoints as well as a traditional 3D reconstruction from off-the-shelf structure-from-motion (SfM) and multi-view stereo (MVS) methods. The 3D reconstruction is usually incorrect for the thin structure; our goal is to segment it out on the correct 3D supporting geometry to improve image based rendering. Multiple factors make this problem hard. The structures are very thin and often lack texture; as a result standard descriptors are ineffective and regularization is difficult. Also, the see-through nature of these objects makes multi-view inference challenging.

Our problem is related to *de-fencing* methods (e.g., [PBCL10]) but their objective is often limited to removing occluding fences, in which case segmentation can be conservative and less precise. The foreground layer can be more precisely estimated [XRLF15], but this requires the small baseline of video sequences.

The key intuition in our work is that we can use multi-view information and the partial 3D reconstruction to estimate segmentation of thin structures in multiple input views. In addition to the images and reconstruction we use as input, a short user interaction is needed to define the supporting geometry and region of interest. The first part of our algorithm is a Markov Random Field (MRF)-based multi-view segmentation, which can handle *multiple layers* of thin structures. We combine appearance cues from color models, median colors and variance, with multi-view consistency constraints on segmentation results. The second part of our solution is an IBR algorithm that allows *free-viewpoint* rendering of multi-layer thin structures. For a given fragment, our renderer interprets alpha values as probabilities to be on a thin structure. These are used to blend weighted colors from the different views.

Our contributions can be summarized as follows:

- A multi-view segmentation algorithm for thin structures supported by simple geometries, that uses multi-view links and color variance to resolve hard ambiguous cases.
- An end-to-end solution to IBR for such *multi-layer* thin structures, including preprocessing to ensure accurate segmentation, and post-processing to generate a clean background.
- A new IBR algorithm that allows *free-viewpoint* navigation of scenes containing these structures.

Our results show significant improvement over previous work, in terms of the identification of the thin structures, the resulting segmentation, and most importantly in the quality of the final image-based rendering in free-viewpoint navigation.

2. Related Work

Our work is related to image *de-fencing*, repetitive structure detection and multi-view segmentation. We also briefly discuss aspects of 3D reconstruction and Image-Based Rendering research related to our work.

2.1. De-fencing and Repetitive Structure Detection

Hays *et al.* [HLEL06] divide research on discovering repeated elements into two extremes: the first focusing on the individual element [LF04] and the other imposing strong structure priors on the general layout of the repeating elements [TTVG01]. Hays *et al.* are the first to automate lattice detection in real images without pre-segmentation. Further improvement is proposed by Park *et al.* [PBCL09] by solving the problem in an MRF setting. More recently, Liu *et al.* [LNS*15] avoid using interest points and apply the Generalized PatchMath algorithm in combination with Particle Belief Propagation to infer the lattice structure. In the case of facades, vanishing lines [WFP10] can be used for plane detection and rectification. In this case dense descriptors are matched not only using repetition but also symmetry. In a multi-view setting, it is possible to detect repetitive elements on more complex surfaces [JTC11] using reconstructed 3D geometry and the images. This however cannot be applied to thin structures in our scenes as there is often no reliable 3D reconstruction.

The problem of image *de-fencing* consists in removing fences from pictures or videos. Liu *et al.* [LBHL08] were the first to propose an automatic method to detect and segment fences in images. Texture based inpainting [CPT04] is used to fill the extracted regions. Fences are found by searching for a lattice that explains the relationship between repeated elements in the image [HLEL06]. This method is further improved [PBCL10] using a multi-view approach for inpainting and a different lattice detection algorithm [PBCL09]. In the case of videos, optical flow is the main cue used to identify fences. A first method is proposed by Mu *et al.* [MLY12] based on motion parallax. Recently, a robust method for obstruction free photography [XRLF15] was proposed to handle occlusion from both fences and windows, generating an alpha matted thin structure layer. Yi *et al* [YWT16] also rely on motion in their fence/non fence segmentation. Finally, Yamashita *et al.* [YMK10] use multi-focus flash/non flash images to identify regions corresponding to objects closer to the camera. Most of these methods assume that the fence is closer to the camera than our typical input. In video-based methods, optical flow estimation is central, and often fails on the wide-baseline input data we have. We show several comparisons to previous methods in Sec. 7.2 and in supplemental material which demonstrate that these methods are not adapted to our goals.

2.2. Multi-View Segmentation

Our work is also related to multi-view segmentation methods that try to identify the foreground object visible in different viewpoints. Some of these methods rely on consistency of the projection of 3D points [KBC12, DFB*13] which is unreliable with thin structures. Other segmentation approaches use constraints from stereo [KSS12] or matching along epipolar lines [CVHC11]. Both strategies are not designed to handle ambiguities due to the often repetitive pattern. In [WFZ02], Wexler et al. estimate two layers from multiple images using a Bayesian framework. Contrary to our work, they assume foreground and background transformations are modeled by planar projections. The resulting Maximum-a-Posteriori estimation requires the definition of a reference view which is not trivial when considering more complex camera setups. Our approach is able to handle complex 3D geometries and multiple layers. Thanks to our multi-view constraints, segmentation is estimated in the original input images for best rendering results.

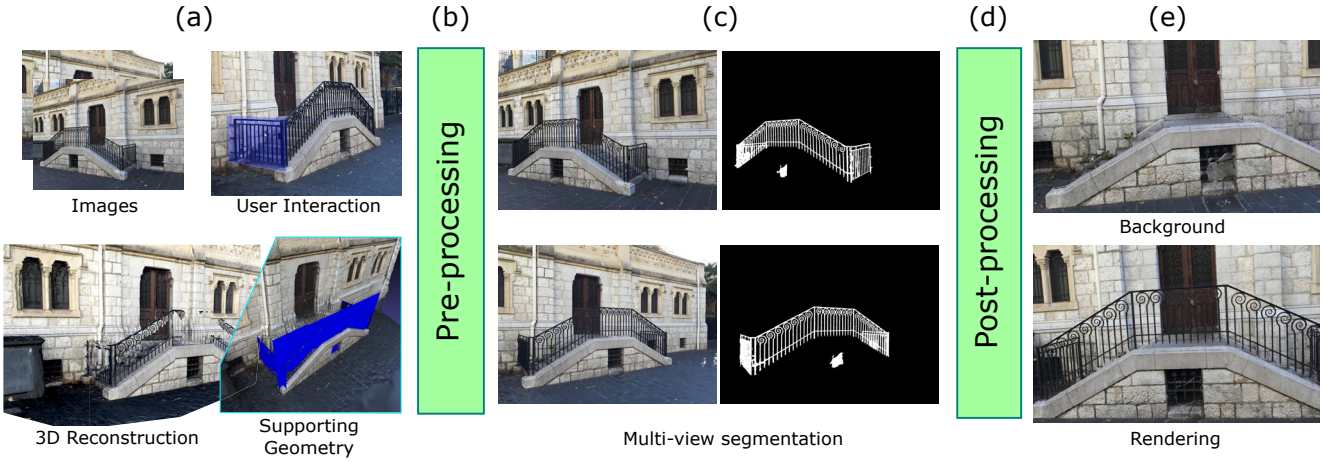

Figure 2: ***End-to-end solution for thin structures in IBR****. (a) After reconstruction and supporting geometries extraction from user interaction, we proceed to (b) pre-processing step to remove spurious geometry. We then perform (c) multi-view,* multi-layer *segmentation followed by (d) post-processing to create "clean" background images and geometry. The result can be used by our multi-layer rendering algorithm (e), allowing* free-viewpoint *navigation.*

2.3. Multi-View Stereo Reconstruction and IBR

Thin structures are also an important limitation for multi-view stereo (MVS) reconstruction methods [GSC*07, FP10, JP11]. Ummenhofer and Brox [UB13] propose a method to reconstruct thin objects which have almost no volume compared to the surface size. Because of the errors in the 3D reconstruction and the normals, the object is not reconstructed (e.g., a sign observed from opposing viewpoints). In a different setup, Oswald *et al.* [OSC14] enforce connectivity constraints and surface genus in temporal 3D reconstruction. In their paper on scene abstraction, Hofer *et al.* [HMB16] use 3D lines to represent 3D scenes. There also has been work specific to the reconstruction of wire structures from images [LCL*17, MMBP14, CH04]; these methods have strong assumptions about the simplicity and/or tubular structure of the objects being reconstructed, and thus do not apply to our context. In general, the hypotheses on input in all of these methods are very different from ours, making them inappropriate for our data.

Initial IBR algorithms did not require geometry [MB95, Che95]. More recent solutions use MVS reconstruction to provide high-quality free-viewpoint navigation when reconstruction works well, e.g., Unstructured Lumigraph (ULR) or more recent methods [EDM*08, LKM14, OCDD15]. The harder case of reflections has been addressed with explicit reflection reconstruction [SKG*12], gradient domain rendering [KLS*13] or stock 3D models [OCDM*16], while vegetation can be handled using oversegmentation and depth synthesis [CDSHD13]. Layered Depth Images [SGHS98] can be used with image data based on an ordering algorithm to allow correct alpha blending; in contrast we interpret alpha values as probabilities to allow specific visibility processing for our multi-layer semi-transparent thin geometries. In recent work the Soft3D algorithm [PZ17] uses a volumetric depth-sweep approach for IBR with a set of clever filtering steps, based on guided filtering and soft visibility, with excellent results. The volumetric nature of this approach means that very fine resolution at unacceptable storage/computation cost would be needed to represent the thin structures we treat. Our rendering algorithm shares some ideas with the soft visibility approach of Soft3D which we discuss in more detail when presenting our method. The central difference is that we work with surface-based 3D (MVS reconstruction and thin supporting geometries) rather than volumes, which allows better visual quality for free-viewpoint navigation far from the input cameras.

3. Overview

Our pipeline is shown in Fig. 2. We start with a set of photographs of the scene together with a 3D reconstruction, estimated using multi-view stereo (e.g., [JP11, Rea16]): We call this the *proxy* geometry and we assume that it completely covers the background of the scene. The 3D model usually represents the thin structure poorly and we provide a user interface to specify the supporting 3D geometry of the thin structure. The user manually specifies 3D points in images, by clicking in 2 images per point. For a plane, 3-4 points are needed and for a cylindrical segment, the user specifies 4-5 points for each of the "upper" and "lower" circles defining the cylindrical segment. The output is a collection of meshes that roughly cover the thin structures (illustrated in Fig. 2.a). This user interaction takes no more than a few minutes for all our examples (please see video).

Our goal is to segment pixels belonging to the thin structure in all the input views (Fig. 2. c). There can be multiple overlapping layers of see-through thin structures. As a result, in addition to separating the thin structures from the background, we also need to segment the thin structures into a distinct set of layers. This makes traditional appearance terms less reliable. Moreover, the segmentation of such structures is hard because the regions are not compact and challenge the balance between area and contour terms in traditional segmentation. Our key observation is that we can leverage the multiple input views to resolve ambiguity present in a given

view. Consider Fig. 3: the structures on the left are very hard to extract as they have colors very similar to the door behind. However, the same structures in the right image have high contrast and can be extracted easily.

Our segmentation algorithm relies on color models and multi-view constraints: for occluded layers, we use median colors [WFZ02] to leverage the multiple views and obtain a color model that is more robust to occlusions. However, for the front-most layer, any inconsistency in colors between the different views is likely to indicate incorrect segmentation and using color variance across viewpoints is more effective. In addition to this, we use multi-view links to help resolve ambiguous cases for segmentation. The segmentation operates layer by layer, from front to back; we will refer to "foreground" as the current front-most thin structure and "background" as the layers behind together with the non-thin parts of the scene. Because we leverage the discrepancy between reprojections to background and foreground layers, we assume that the thin structure has not been reconstructed in the 3D proxy. To ensure this, we remove geometry in the close neighborhood of the supporting geometry during a preprocess step.

After multi-view segmentation, we refine the segmentation using off-the-shelf alpha matting [HRR*11], whereas the background regions occluded by the thin structures are filled using the median images and inpainting. The resulting images are then used to create a "thin-structure-free" 3D reconstruction of the background. These correspond to the post-processing in Fig. 2(d). Details for pre- and post-processing are provided in Sec. 5.

We introduce a new rendering algorithm that handles multi-view alpha mattes, by interpreting alpha values as conditional probabilities that a fragment contains a thin structure. We estimate the overall alpha value using Bayes rule. Rendering involves a depth-peeling algorithm on the supporting geometries of the thin structures, rendered after a first ULR pass of the clean background (Fig. 2. e). Our results on scenes with thin structures show significantly better quality than previous methods, especially for the free-viewpoint navigation far from the input cameras.

4. Multi-view Segmentation

To resolve the difficult multi-layer segmentation problem, we use multi-view color and geometry information.

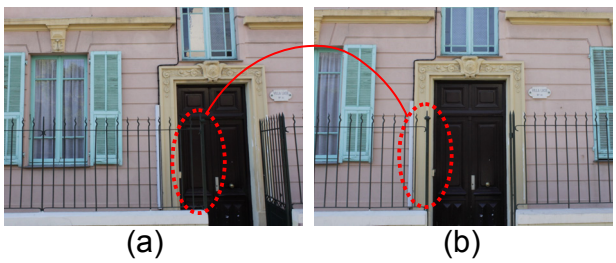

Figure 3: ***Benefits of a multi-view approach.*** *Segmentation of the ambiguous area on the left can be resolved using the view on the right.*

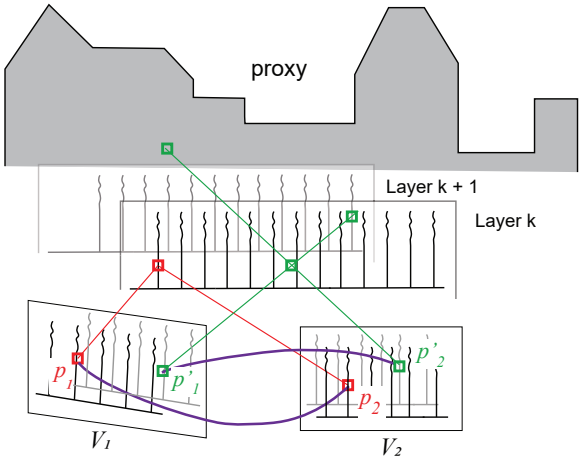

Figure 4: ***Multi-view links at iteration*** *k. We create multi-view links by connecting two pixels which correspond to the same 3D point on the layer k (p_1 linked to p_2 and p'_1 linked to p'_2). When the 3D point really corresponds to a thin structure ($p_1 \leftrightarrow p_2$), we have linked together two pixels from the layer k. When the 3D point is in-between the thin structure ($p'_1 \leftrightarrow p'_2$), we do not know which layer corresponds to which pixel: p'_1 is in layer $k+1$, while p'_2 is on the proxy but we can infer that the two pixels are on layers $> k$.*

4.1. Multi-Layer Segmentation

Our approach handles multiple layers of thin structures, e.g, the corner of the staircase in Fig. 2. For each pixel p in each input image, we create a sorted list of N_p front-to-back depth candidates d_k by ray-casting the proxy and the thin structure supporting geometries. The last depth candidate d_{N_p} per pixel is always the proxy depth because we assume that the proxy is opaque, so we stop ray-casting as soon as it is hit.

We denote by $\mathcal{P}$ the set of all pixels p for which $N_p \geq 2$. It corresponds to all pixels that need to be segmented because they have at least one depth candidate in addition to the proxy depth. Solving the general multi-layer segmentation problem is equivalent to assigning the correct depth to each pixel, which we model as finding a labelling that will minimize an energy function defined by multi-view information on color and geometry.

To solve the multi-label problem, we decompose it into a sequence of binary label problems, in an alpha-expansion-like fashion [BVZ01], considering depth labels from front-to-back. For each iteration k, we select all the pixels with current segmentation $\geq k$, and we want to find out how much we can expand the depth layer k. The two labels for the sub binary problems are "depth layer is k" and "depth layer is $> k$", i.e., "foreground" and "background" respectively.

To see the justification for this decomposition, consider Fig. 4: labels of pixels associated through the current depth layer k are linked. They are either both at the correct depth layer k, which is the case for pixels p_1 and p_2, or both at a depth layer $> k$ in the case of pixels p'_1 and p'_2. In this case, it is not possible to associate a specific layer $> k$ to the pixels p'_1 and p'_2 (see Fig. 4).

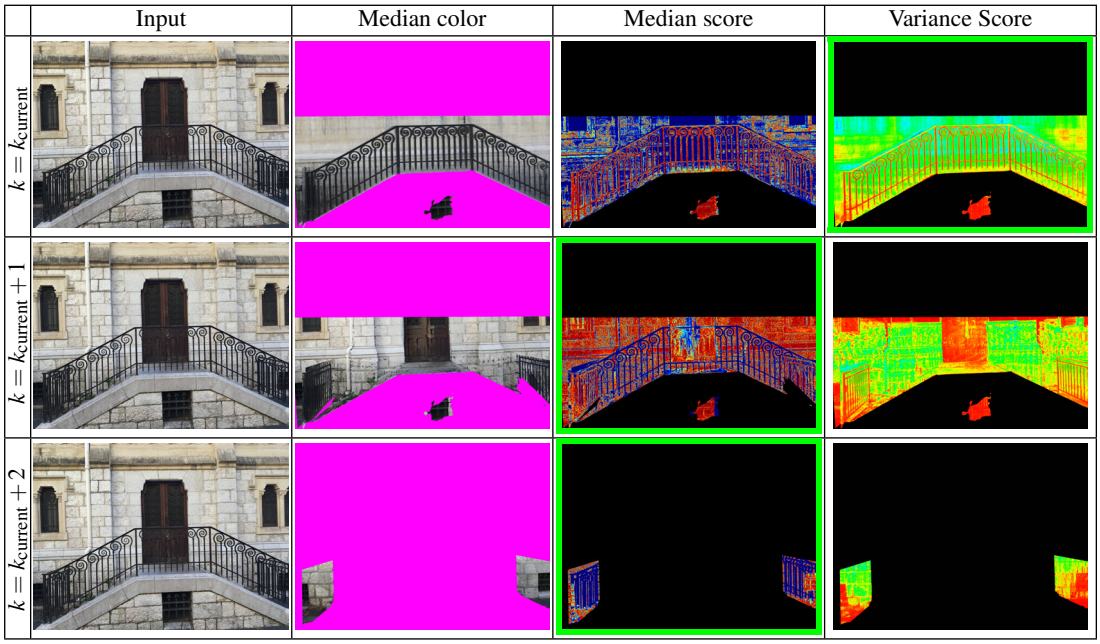

Figure 5: ***Advantage of using median or variance information for each depth layer for a given current depth layer****. Warm colors in the score columns indicate high likelihood for this layer to give the correct depth. For the current layer, the variances give cleaner information. For the layers beyond, the medians remove outliers while the variance score, more sensitive to occlusions, is low on the wall. For each layer, the most informative score is outlined in green.*

We denote $\mathcal{K}$ the geometry corresponding the depth layer k. We use the term *background geometry* to refer to the geometry corresponding to all depth layers $> k$, i.e., thin structures behind the current layer or proxy of the scene. We iteratively solve these binary problems for $k = 1$ to $k = \max_{p \in \mathcal{P}} N_p - 1$.

4.2. General Formulation

We can now express the segmentation as a binary labeling problem for each pixel, minimizing an MRF energy function. We start with standard color and smoothness terms, and use two terms that exploit multi-view cues. The first provides *a multi-view color prediction* of a foreground vs. background pixel given the reprojections from other views, while the second term *links the label* of pixels from *different viewpoints* if they correspond to the same point on the foreground layer $\mathcal{K}$.

We next describe the graphcut segmentation terms before introducing our new terms in more detail.

Color Model. We compute prior probabilities for each pixel for being in the thin structure or in the proxy: $P^{\text{color}}_{\text{thin}}$ and $P^{\text{color}}_{\text{proxy}}$ respectively. These are estimated from global per view appearance models. In our case, we use histograms. We cluster all the colors in all input images and for a color c, we note $C(c)$ its associated cluster, where this cluster defines the corresponding bin in the histogram. We note $\mathcal{P}^{\mathcal{I}}_{\text{thin}}$ the set of pixels of image $\mathcal{I}$ currently segmented as one of the thin structures and $\mathcal{P}^{\mathcal{I}}_{\text{proxy}}$ the set of all input pixels not in $\mathcal{P}^{\mathcal{I}}_{\text{thin}}$. Probabilities for a pixel p from image $\mathcal{I}$ with color c_p are then defined as:

$$
\begin{aligned}
P^{\text{color}}_{\text{thin}}(p) &= \frac{\#\{q \in \mathcal{P}^{\mathcal{I}}_{\text{thin}}, C(c_q) = C(c_p)\}}{\#\{q \in \mathcal{P}^{\mathcal{I}}_{\text{thin}}\}} \\
P^{\text{color}}_{\text{proxy}}(p) &= \frac{\#\{q \in \mathcal{P}^{\mathcal{I}}_{\text{proxy}}, C(c_q) = C(c_p)\}}{\#\{q \in \mathcal{P}^{\mathcal{I}}_{\text{proxy}}\},}
\end{aligned}
\tag{1}
$$

where # is set cardinality. Finally, the probabilities P^{color}_k to be part of the layer k based on the color models are given by:

$$
P^{\text{color}}_k(p) = \begin{cases} P^{\text{color}}_{\text{proxy}}(p) & \text{if } k = N_p, \\ P^{\text{color}}_{\text{thin}}(p) & \text{otherwise.} \end{cases}
\tag{2}
$$

Smoothness term. We use a standard pairwise contrast sensitive energy term, with a 4-neighbor connectivity. Specifically:

$$
E_{\mathcal{N}}(p,q) = \begin{cases} \frac{1}{1+||c_p - c_q||} & \text{if } l_p \neq l_q, \\ 0 & \text{otherwise.} \end{cases}
\tag{3}
$$

with $||c_p - c_q||$ the L_2 norm of the color difference.

4.3. Multi-View Color Term

We leverage multi-view information to determine the probability of being foreground or background by *reprojection*, in the spirit of multi-view stereo [SCD*06]. Standard reprojection error approaches are not sufficiently robust in our challenging context, in particular because of occlusion. Instead we compare observed pixels to the *median* of the reprojections from multiple views [WFZ02] and introduce *variance* to allow more robust results.

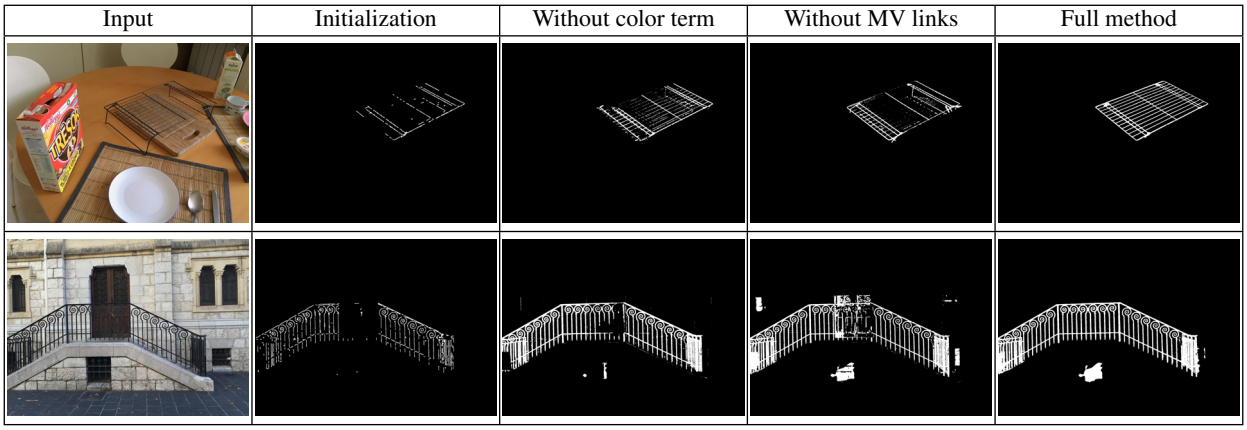

Figure 6: ***Advantage of using a multi-view approach for segmentation.*** *The MV links allow us not only to retrieve missing segmentation because of a similar background but also to remove wrong segmentation because of possibly misleading unary terms.*

For a 3D point in the current depth layer k_{current}, since we have an approximate segmentation of the previous depth layers, we can compute a reliable visibility information from input views. So with the hypothesis of low specularity, a high color variance in the re-projections of the 3D point into the input views is a strong cue of an incorrect depth. On the other hand, for a 3D point in a depth layer $k > k_{\text{current}}$, we do not have reliable visibility information, so even with a correct depth, a high color variance could simply be a consequence of occlusions. However, if we make the assumption that since the foregrounds are thin, a background point will not be occluded in the majority of the views, we can use the median color across views to remove occlusion outliers.

For a given view, we use the 3D information from layer k to project into the other views and collect color samples. The variance map $\sigma_{\mathcal{I}}$ defines the variances of these samples for each pixel p. The median image $\mathcal{M}$ is obtained by sorting the samples by luminance, and keeping the median color value for each pixel.

The likelihood for a pixel p to be part of depth layer k is

$$\mathcal{L}_k(p) = \begin{cases} \mathcal{G}(\sigma_{\mathcal{I}}(p), 0, \sigma_{\text{ref}}) & \text{if } k = k_{\text{current}}, \\ \mathcal{G}(\mathcal{I}(p), \mathcal{M}_k(p), \mathcal{I}_{\text{ref}}) & \text{if } k > k_{\text{current}}. \end{cases} \tag{4}$$

with $\mathcal{G}$ indicating a Gaussian distribution. In the case of the current depth layer k_{current}, a small variance in the color samples is likely to indicate correct depth assignment. In the case of other depth layers $k > k_{\text{current}}$, using the median color map $\mathcal{M}_k$ is more robust to the occasional occlusions. We show examples of the median images in Fig. 5. In the case of the front-most layer ($k = k_{\text{current}}$), using the variance map gives more precise results whereas the median colors are more effective on layers further away ($k > k_{\text{current}}$).

Moreover, since the layers occlude each other, we introduce a prior weight λ_k to take in account that the segmentation of layer k becomes more likely as it approaches the front (excluding the proxy which is prominent). Given that the layer $k = N_p$ corresponds to the proxy, we model this by a Bernoulli trial of parameter 0.6 such that $\lambda_{\text{proxy}} = \lambda_{N_p} > \lambda_1 > \lambda_2 > \cdots > \lambda_{N_p-1}$.

Therefore the probabilities P_k^{mv} to be part of the layer k ($k \geq k_{\text{current}}$), based on the multi-view information are given by:

$$P_k^{\text{mv}}(p) = \frac{\lambda_k \cdot \mathcal{L}_k(p)}{\sum_{k' \geq k_{\text{current}}} \lambda_{k'} \cdot \mathcal{L}_{k'}(p)} \tag{5}$$

4.4. Multi-View Links

We introduce multi-view links to enforce consistency between views, transferring more reliable segmentations to views with harder configurations. We reproject each pixel into neighboring images with depth $\mathcal{K}$ and we create a link for each projection if its current segmentation is $\geq k$. We do this since if the reprojection falls on a pixel with current segmentation $< k$, we do not want to change this decision; see illustration in Fig. 4, To limit issues due to coordinate rounding, links are created only if the backprojection in the opposite direction is on the same pixel.

$E_L(p,q)$ is the energy term corresponding to these multi-view links. It is created between all pairs of views (i, j), linking a pixel p from i with a pixel q from j through reprojection:

$$E_L(p,q) = \begin{cases} \lambda & \text{if } l_p \neq l_q, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

These links allow the propagation of segmentation results from views where color models are more discriminatory to images where the difference between foreground and background appearance is ambiguous. In all our results, $\lambda = 1.0$.

4.5. Unary Energy Terms

The probabilities described above give information about a layer k. The unary cost associated to a layer k is given by :

$$\mathcal{U}_k(p) = -\log\left(\lambda_{\text{color}} \cdot P_k^{\text{color}}(p) + \lambda_{\text{mv}} \cdot P_k^{\text{mv}}(p)\right) \tag{7}$$

Terms λ_{color} and λ_{mv} are mixing coefficients between the global and the multi-view appearance models, where $\lambda_{\text{mv}} = 0.4$ and $\lambda_{\text{color}} = 1 - \lambda_{\text{mv}}$. We now need to compute the unary terms with

respect to the labels "foreground" and "background". For the foreground label "k" the unary term is simply the unary cost associated to that depth layer. For the background label "$> k$" the unary term is the minimum of all the unary costs associated to the depth layers $k+1, k+2, \cdots$. We take the minimum to be as conservative as possible when assigning the label k, because if k is chosen, the pixel will not appear in the following binary segmentation problems as we solve front-to-back. Therefore the unary energy term is given by:

$$E_U(p) = \begin{cases} \mathcal{U}_k(p) & \text{if } l_p = \text{"foreground"}, \\ \min\limits_{k'>k} \mathcal{U}_{k'}(p) & \text{if } l_p = \text{"background"}. \end{cases} \quad (8)$$

4.6. Final Energy Formulation

$E_N(p,q)$ and $E_L(p,q)$ are pairwise terms used to enforce coherent segmentation respectively at image and multi-view level.

If we note by $\mathcal{N}$ all the pairs of neighboring pixels, and by $\mathcal{L}$ all the multi-view links, the final energy optimized by the MRF is as follows:

$$E = \sum_{p \in \mathcal{P}} E_U(p) + \sum_{(p,q) \in \mathcal{N}} E_N(p,q) + \sum_{(p,q) \in \mathcal{L}} E_L(p,q) \quad (9)$$

In Fig. 6 we illustrate the effect of the unary terms and the multi-view consistency term $E_L(p,q)$. Using only classic color models for foreground and background is sensitive to any similarity of appearance between these two parts. Adding the multi-view color term helps to take into account the other viewpoints and we clearly see that some regions that were erroneously part of the foreground model, are now less likely to be coherent in multi-view. We also see why multi-view links allow the propagation of good segmentation results across viewpoints.

5. Pre- and Post-Processing

Achieving high quality segmentation requires a few pre-processing steps, which we describe below. We also explain the matting and background image creation steps, required for rendering. Recall that the input to these steps is the multi-view stereo reconstruction with calibrated cameras, and the user-defined thin structure geometry.

5.1. Pre-processing

Our multi-view segmentation approach assumes that the thin structures are not reconstructed. In some cases, modern reconstruction algorithms create spurious geometry in the thin structure supporting surface $\mathcal{S}$, typically filling the space between them or even reconstructing small parts of the structure. To allow high-quality segmentation, we need to remove this reconstruction, bringing our input data into the canonical form expected by the segmentation.

Removing reconstruction in the thin structure geometry. To identify the vertices of the proxy that comes from superfluous reconstruction (e.g., false surfaces between thin structured), we first compute a smoothed histogram of the log point-to-mesh L2 distance between the proxy vertices and the thin structure geometry. We then remove all the vertices for which the log-distance is smaller than the value associated to the first valley (see Fig. 7). Such removal can reveal holes in the proxy that need to be filled in order to have a depth available behind the thin structure geometry. We fill those holes by fitting local planes.

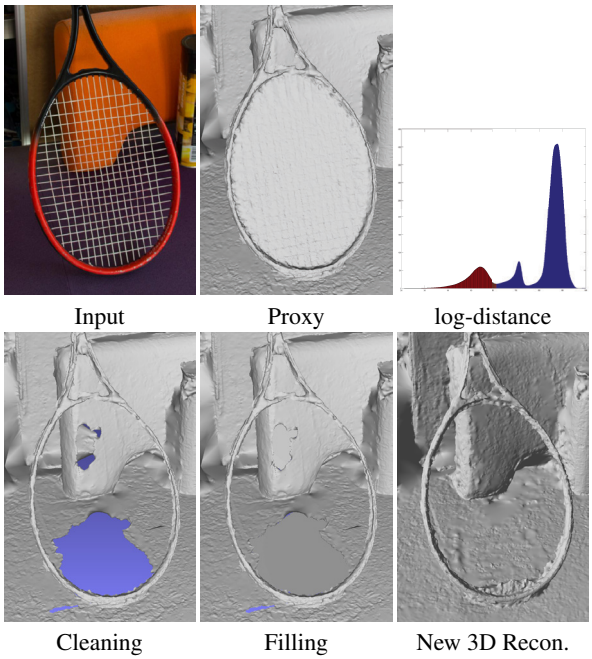

Figure 7: *Result of the depth pre-processing for regions with spurious reconstructed geometry in the geometry of thin structures. The histogram shows the log distances distribution between the supporting geometry and the proxy vertices. The closest vertices (red bins) are removed during the cleaning step.*

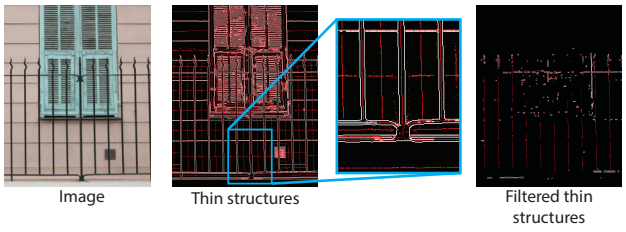

Figure 8: *Thin structure candidates for a given input image. Thin structures are marked in red, between detected edges in white. We filter these thin structure candidates using multi-view information. To initialize the multi-view segmentation, we select thin structure candidates p that have $P_k^{mv}(p) > 0.9$*

Detection of Thin Structure candidates. We obtain first candidate points for thin structures by detecting edges in the images [Can86]. Candidate points are generated between the edges. This is illustrated in Fig. 8: the edges detected are in white, and the generated candidates points are in red. Since a large number of such candidates can be generated, we filter the result using multi-view information. More specifically, we initialize a thin structure candidate to

be at layer k if $P_k^{\text{mv}}(p) > 0.9$ for any $k < N_p$. Every other pixel is initialized with the back-most layer.

5.2. Post-processing

In post-process, we need to extract the foreground and an alpha matte of the thin structures, create the background image corresponding to each input view and the clean background geometry. Before matting, we remove small spurious outliers, by removing connected components with size less than 0.2% of the image size.

Alpha Matting. The tri-map for the alpha matting step is obtained by first dilating the foreground label maps to create the uncertain region and eroding to determine the foreground. The resulting segmentations for two images of one of our test scenes are provided as supplemental material. The accuracy of the tri-maps is important to the success of the subsequent steps, but also depends on the matting algorithm used. We use the approach of [HRR*11] for matting, which provides adequate alpha mattes for rendering.

Background Generation. We use the median images, updated with the visibility information from the segmentation, as background in the regions labelled as thin structures, with a Poisson editing step to correct for small differences in color levels. A small number of pixels are black in these images, corresponding to regions occluded by the thin structures in all images; we apply standard single-image inpainting to fill these holes using a variant of PatchMatch with the "occurence" term of Kaspar et al. [KNL*15]. Once we have generated these color-balanced background images, we perform a fresh 3D reconstruction step with the thin structures removed. This step improves the quality of the background mesh significantly (see video).

6. Rendering

Our rendering algorithm is based on the Unstructured Lumigraph (ULR) [BBMC01] with soft visibility [EDM*08], operating on a per-pixel basis. Standard ULR precomputes a depth buffer for each input view. During rendering, a first pass renders the depth to the frame buffer, and for each fragment, we perform a depth test for each input view with the precomputed depth. If the test fails the fragment is discarded (Fig. 9(a)). If it succeeds, the ULR weights w_i are computed [BBMC01] expressing the match in angle and distance between the novel and input views, and used to blend the corresponding colors from the input images.

Our scenes are more complex. The depth map for each input view is computed using the background geometry, and then augmented with the information provided by the segmentation. Each pixel on a thin structure has a depth corresponding to the supporting geometry layer determined by the segmentation, and an alpha value computed by the matting process (Fig. 9(b)). There are two main issues: first, the supporting geometry of the thin structure (dashed lines in Fig. 9(b)-(d)) is semi-transparent, and thus requires a specific rendering algorithm and second, we need to define how to combine the view-dependent alpha values in the novel view.

For the first issue, we adapt the depth peeling algorithm [Eve01], run in back-to-front order. This consists in progressively rendering each layer of depth with a "less-than" depth test and alpha blending with the previous layer. We first render the clean background proxy using standard ULR, then perform depth peeling using the supporting geometries.

For the second issue we propose an interpretation of α values as conditional probabilities of having a thin structure along the viewing ray of a pixel in each input view. $P(\text{thin}_f|V_i)$ is the conditional probability that a fragment f is on a thin structure, given view V_i. We thus assume $\alpha_i = P(\text{thin}_f|V_i)$; we interpret ULR weights as a prior probability of each view V_i:

$$P(V_i) = \frac{w_i}{\sum_k w_k}. \tag{10}$$

We will apply Bayes rule to compute the overall probability of a given fragment being on a thin structure, at a given layer of the depth peeling algorithm, which can be seen as a multi-view filtering of the segmentation results. Then we use this probability as the α_f value for blending the weighted ULR colors from the input views for fragment f (see Algorithm 1 for the details):

$$\alpha_f = P(\text{thin}_f) = \sum_i P(\text{thin}_f|V_i)P(V_i) = \frac{\sum_i \alpha_i w_i}{\sum_i w_i} \tag{11}$$

There are two cases that need to be treated as specific depth tests. First, if a fragment of a thin structure from another input view V_j is in front of the current thin structure fragment f, then V_j is ignored by setting $w_j = 0$ (e.g., Fig. 9(c), V_1 is ignored), since it cannot provide information about a fragment behind it. The second case is if there is a thin structure at a depth behind the fragment f in an input view. Since the segmentation has placed it behind f, the conditional probability that the thin structure is at the depth of f is 0; we thus set $P(\text{thin}_f|V_i) = \alpha_i = 0$. However, we include the input view in the blend of colors to be conservative. This approach is related to the consensus voting approach in Soft3D [PZ17]. Since we have surface-based geometry, instead of performing volumetric accumulation, we can directly used the probabilities computed to make a decision for each fragment, avoiding visual artifacts far from the input cameras.

ALGORITHM 1: Multi-Depth Thin Structure Rendering Algorithm.

Input: Images with depth, background and thin geometry
Result: Novel View
Render clean background with standard ULR into C_{current};
for *each depth peel layer, back to front* **do**
  Render alpha with depth tests in α_f;
  Render color with ULR weights in C_f;
  Blend with previous layers : $C_{\text{current}} \leftarrow C_f \cdot \alpha_f + (1-\alpha_f) \cdot C_{\text{current}}$;

7. Results and Comparisons

We first show results of our method on a variety of scenes. We then provide comparisons with related work, and finally discuss limitations. When judging comparisons, one should consider that our results benefit from the user-defined thin geometry, but this information only could not be directly used by these previous algorithms.

7.1. Results

We ran our approach on an Xeon E5-2650 PC. For each scene we took between 10 and 30 photos. The outdoors scenes were taken

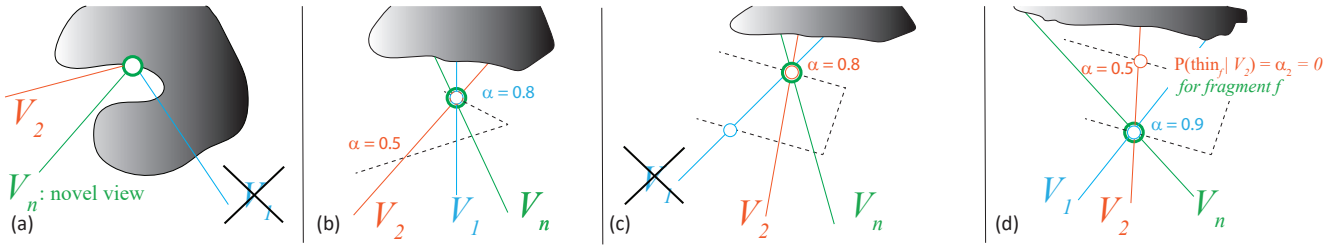

Figure 9: *(a) Traditional ULR rendering: For fragment f (green circle) we ignore view V_1. (b) In our case, we have multiple values of* α *from different views, that blend with the background. These are interpreted as probabilities of being a thin structure at this depth, thus in this example $P(thin_f|V_1) = \alpha_1 = 0.8$ and $P(thin_f|V_2) = \alpha_2 = 0.5$. (c) If for an input view the depth map is in front of the current fragment, the view (V_1 here) is ignored as the view gives no information about the fragment. (d) If for an input view, the depth map is behind the current fragment, the view (V_2 here) estimates there is no thin structure at the fragment location with $P(thin_f|V_2) = 0$.*

with a DSLR camera, and the indoors with an iPhone 6S. We use [Rea16] for camera calibration and 3D reconstruction. Note that we assume that a background depth is reconstructed for all pixels in an image; To treat a scene with a sky background for example, a bounding box could be fit to the scene to provide "far depth" for sky pixels. The user specifies the supporting geometry by providing correspondences points in a multi-view interface to fit a basic 3D primitive and then manipulates the primitive control points in a 3D window; Our current interface handles planar segments and cylinders. Please see the video for an example interaction session.

The whole offline pipeline (excluding reconstruction steps) took in the order of 5 minutes for the smaller scenes (∼ 10 images) to 20 minutes for the larger scenes (∼30 images). The rendering runs at 60 fps on a 1080p display with all datasets using a GTX 1080.

Our approach allows us to capture complex shapes which are not necessarily purely repetitive, for example in the "Stairs" scene. We provide the multi-view segmentations and the mattes for all our datasets in supplemental material. Small artifacts in the segmentation are often alleviated by the blending step of IBR. We show the results of our inpainting step and rendering from novel viewpoints in Fig. 10. The Stairs and Rolland scenes contain multiple layers of thin structure depth, and the Balcony scene is a cylinder with multiple depths. The supporting geometry could in theory have any form, as long as it can be reasonably represented by a mesh.

7.2. Comparisons

We present comparisons on rendering and segmentation. For rendering, please see the accompanying video, the improvement over of previous methods is much more evident during interactive free-viewpoint navigation.

Rendering. Our approach allows free-viewpoint navigation to regions quite far from the original cameras. We thus focus our comparisons on other methods that allow interactive free-viewpoint navigation, in particular Unstructured Lumigraph Rendering (ULR) [BBMC01] and Bayesian IBR [OCDD15] (which subsumes [CDSHD13]). For these comparisons we use our re-implementation of ULR, which in contrast to the original version, uses *per-pixel* weights based on the geometric proxy from MVS reconstruction, and the original implementation of Bayesian IBR [OCDD15]. We show results for several scenes and novel viewpoints, far from the input cameras. Our method greatly improves over previous methods, since those methods incorrectly project thin structures onto the background geometry.

In addition to these free-viewpoint methods we compare to the recent Soft3D IBR method [PZ17]. We use an reimplementation of the complete Soft3D method provided by P. Hedman of UCL †. As can be seen in the video, Soft3D produces high quality results for certain capture configurations (e.g., all cameras in a plane) and when interpolating camera positions. When moving away from the input cameras, blurring artifacts appear due to the depth discretization. These are even more visible for thin structures. Note also that rendering a single image with Soft3D takes seconds, compared to the real-time behavior of our method.

Segmentation. To our knowledge, no previous method has addressed thin structures for image-based rendering. As a result, we compare with the most closely related methods, which are not tuned for our data. For completeness, we also compare to previous lattice-detection methods in supplemental material. The results show that such methods either fail to find parts or even most of the thin structures, or sometimes only partially succeed for some but not all of the images in the multi-view dataset. This is unsuprising given our richer input (camera calibration, 3D reconstruction and user-defined thin geometry); The comparison is provided to show that these methods cannot solve our problem.

We also compare to automatic fence segmentation [YWT16] in Fig. 11. The authors kindly ran their code on our dataset, however since our data are not continuous videoframes their method cannot perform spatio-temporal refinements based on frame-by-frame optical flow. Thus the results shown are the "initial segmentation" described in the paper. If the input requirements for this method are respected, the results would be better. Similarly, we compare to the method of Xue et al. [XRLF15] run on a crop of the input images. Again this method expects a smaller baseline between images, and thus cannot be used for our target scenes.

† We provide a comparison of our results for the Soft3D algorithm and the original implementation on the Museum scene [CDSHD13] as an additional video. The general image quality is similar, even though the original implementation is slightly sharper.

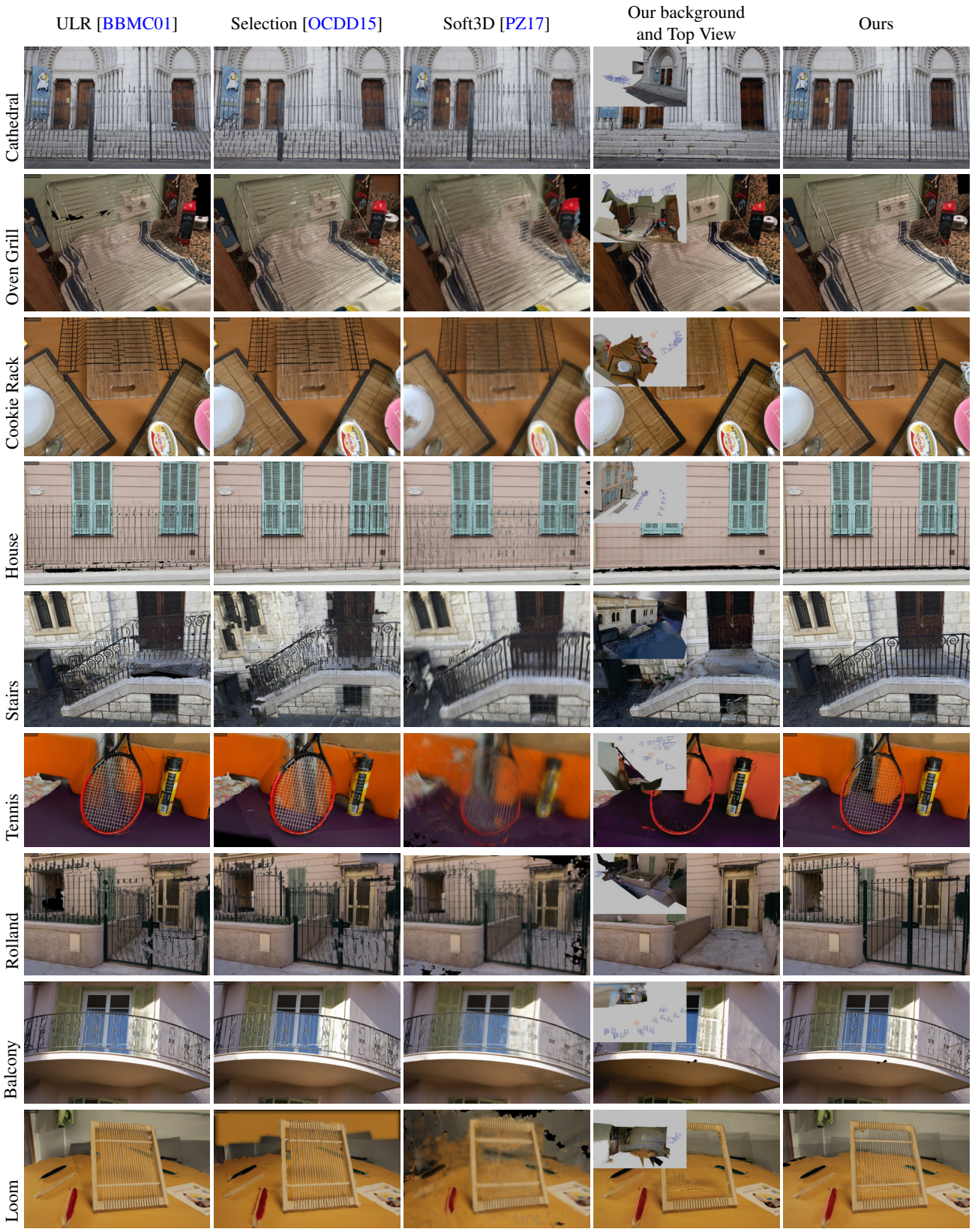

Figure 10: *Rendering results for 9 different scenes from novel views far from the input cameras. From left to right: rendering results for [OCDD15], our improved version of [BBMC01], a re-implementation of Soft3D [PZ17], the inpainted background and the top view showing the novel camera, and our result.*

Video de-fencing [YWT16]

Cathedral House

Figure 11: Results of video de-fencing method. *This is the result obtained on* House *and* Basilic *datasets with a method based on motion estimation [YWT16]. In a wide baseline setup, it is unlikely to compute reliable matches between consecutive images.*

Obstruction free photography [XRLF15]

Background Foreground

Figure 12: Comparison with obstruction free photography [XRLF15]. *Results on Cathedral dataset: left image is the background layer and right image is the foreground layer. Optical flow based methods are unlikely to handle wide baseline configurations.*

7.3. Limitations and discussion

Our method fails when the background has the same color as the foreground in all views. Our simple thin geometries are implicitly assumed to be infinitely thin; this works well for most cases, but becomes problematic at extreme grazing angles such as Fig, 13(a). The ability to detect thin structures also depends on the resolution of the images with respect to the thickness of the structures. When the structures are too thin, our approach can fail, as seen in Fig. 13.

Figure 13: *(a) At extreme grazing angles, the infinitely thin assumption is insufficient. (b) Very small structures in input images can be missed by our approach (see rendering in (c)).*

In all our examples, the supporting surfaces produced were accurate enough to produce good quality results. Consequently, we did not perform in-depth robustness analysis. Very imprecise user input could result in inaccuracies for the supporting geometry, affecting the segmentation quality because of the larger number of incorrect multi-view links. For such input, the surface estimation could be refined between segmentation steps, e.g., using intermediate segmentation results to find dense correspondences. Automating the detection of supporting geometries could be achieved with learning-based solutions. Recent high-quality segmentation deep learning networks (e.g., [LMSR17]) could potentially be adapted and trained to identify thin structures, although generating the training data is a challenge. Similarly, deep learning has been used to find parameters of simple geometric structures [NGDA*16]. With suitable knowledge of semantics and the correspondence between supporting primitive type and thin structure, such an approach could be developed to allow primitive fitting based on images, potentially combined with an optimization step [OCDM*16]. Another interesting challenge is the interaction of thin structures and vegetation which renders the problem much harder.

8. Conclusions and Future Work

We have presented a new method to treat the hard problem of thin structures for image-based rendering. The central component of our approach is a segmentation algorithm which exploits multi-view information and 3D reconstruction to provide segmentation of multi-layer thin structures. Our end-to-end solution allows us to produce results showing significant improvement over previous methods in terms of rendering quality, allowing common scenes with thin structured to be used in a free-viewpoint IBR context for the first time.

The key aspect of our work is to leverage multi-view information to overcome the ambiguities in the hard case of thin structure segmentation. Using other priors as discussed above will allow this approach to generalize to other structures in terms of complex geometry and varying frequencies of the repetitive patterns.

9. Acknowledgements

This research was partially funded by the doctoral fellowship of the Region Provence Alpes Cote d'Azur, ANR project SEMAPOLIS (ANR-13-CORD-0003), the EU Horizon 2020 grant No. 727188 EMOTIVE, http://www.emotiveproject.eu/, and by generous donations from Technicolor and Adobe. The authors thank P. Hedman for his Soft3D implementation, J. Philip for mesh alignment code and S. Rodriguez for his precious help finalizing the paper.

References

[BBMC01] BUEHLER C., BOSSE M., MCMILLAN L., COHEN S. G. M.: Unstructured lumigraph rendering. In *SIGGRAPH* (2001). 8, 9, 10

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI 23*, 11 (2001), 1222–1239. 4

[Can86] CANNY J.: A computational approach to edge detection. *IEEE Trans. PAMI*, 6 (1986), 679–698. 7

[CDSHD13] CHAURASIA G., DUCHENE S., SORKINE-HORNUNG O., DRETTAKIS G.: Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. on Graphics (TOG) 32*, 3 (2013), 30. 1, 3, 9

[CH04] CARMICHAEL O., HEBERT M.: Shape-based recognition of wiry objects. *IEEE Trans. PAMI 26*, 12 (2004), 1537–1552. 3

[Che95] CHEN S. E.: Quicktime vr: An image-based approach to virtual environment navigation. In *SIGGRAPH '95* (1995), ACM, pp. 29–38. 3

[CPT04] CRIMINISI A., PEREZ P., TOYAMA K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. on image processing 13*, 9 (2004), 1200–1212. 2

[CVHC11] CAMPBELL N. D., VOGIATZIS G., HERNÁNDEZ C., CIPOLLA R.: Automatic object segmentation from calibrated images. In *Visual Media Production (CVMP), 2011 Conference for* (2011), IEEE, pp. 126–137. 2

[DFB*13] DJELOUAH A., FRANCO J.-S., BOYER E., LE CLERC F., PÉREZ P.: Multi-view object segmentation in space and time. In *ICCV* (2013), pp. 2640–2647. 2

[EDM*08] EISEMANN M., DECKER B. D., MAGNOR M., BEKAERT P., DE AGUIAR E., AHMED N., THEOBALT C., SELLENT A.: Floating textures. *Comp. Graph. Forum* (2008). 3, 8

[Eve01] EVERITT C.: Interactive order-independent transparency. 8

[FP10] FURUKAWA Y., PONCE J.: Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. PAMI* (2010). 3

[GSC*07] GOESELE M., SNAVELY N., CURLESS B., HOPPE H., SEITZ S. M.: Multi-view stereo for community photo collections. In *ICCV* (2007). 3

[HLEL06] HAYS J., LEORDEANU M., EFROS A. A., LIU Y.: Discovering texture regularity as a higher-order correspondence problem. In *ECCV* (2006), Springer, pp. 522–535. 2

[HMB16] HOFER M., MAURER M., BISCHOF H.: Efficient 3d scene abstraction using line segments. *Computer Vision and Image Understanding* (2016). 3

[HRR*11] HE K., RHEMANN C., ROTHER C., TANG X., SUN J.: A global sampling method for alpha matting. In *CVPR* (2011), IEEE, pp. 2049–2056. 4, 8

[JP11] JANCOSEK M., PAJDLA T.: Multi-view reconstruction preserving weakly-supported surfaces. In *CVPR* (2011), IEEE, pp. 3121–3128. 3

[JTC11] JIANG N., TAN P., CHEONG L.-F.: Multi-view repetitive structure detection. In *2011 Int. Conference on Computer Vision* (2011), IEEE, pp. 535–542. 2

[KBC12] KOLEV K., BROX T., CREMERS D.: Fast joint estimation of silhouettes and dense 3d geometry from multiple images. *IEEE Trans. PAMI 34*, 3 (2012), 493–505. 2

[KLS*13] KOPF J., LANGGUTH F., SCHARSTEIN D., SZELISKI R., GOESELE M.: Image-based rendering in the gradient domain. *ACM Trans. on Graphics (TOG) 32*, 6 (2013), 199. 1, 3

[KNL*15] KASPAR A., NEUBERT B., LISCHINSKI D., PAULY M., KOPF J.: Self Tuning Texture Optimization. *Computer Graphics Forum* (2015). 8

[KSS12] KOWDLE A., SINHA S. N., SZELISKI R.: Multiple view object cosegmentation using appearance and stereo cues. In *ECCV* (2012), Springer, pp. 789–803. 2

[LBHL08] LIU Y., BELKINA T., HAYS J. H., LUBLINERMAN R.: Image de-fencing. In *CVPR* (June 2008), pp. 1–8. 2

[LCL*17] LIU L., CEYLAN D., LIN C., WANG W., MITRA N. J.: Image-based reconstruction of wire art. *ACM SIGGRAPH 2017* (2017). 3

[LF04] LOBAY A., FORSYTH D. A.: Recovering shape and irradiance maps from rich dense texton fields. In *CVPR* (2004), vol. 1, IEEE, pp. I–400. 2

[LKM14] LIPSKI C., KLOSE F., MAGNOR M.: Correspondence and depth-image based rendering: a hybrid approach for free-viewpoint video. *IEEE T-CSVT* (2014). 1, 3

[LMSR17] LIN G., MILAN A., SHEN C., REID I.: RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR* (July 2017). 11

[LNS*15] LIU S., NG T.-T., SUNKAVALLI K., DO M. N., SHECHTMAN E., CARR N.: Patchmatch-based automatic lattice detection for near-regular textures. In *ICCV* (2015), pp. 181–189. 2

[MB95] MCMILLAN L., BISHOP G.: Plenoptic modeling: An image-based rendering system. In *SIGGRAPH '95* (1995), ACM, pp. 39–46. 3

[MLY12] MU Y., LIU W., YAN S.: Video de-fencing. *arXiv preprint arXiv:1210.2388* (2012). 2

[MMBP14] MARTIN T., MONTES J., BAZIN J.-C., POPA T.: Topology-aware reconstruction of thin tubular structures. In *SIGGRAPH Asia 2014 Technical Briefs* (2014), ACM, p. 12. 3

[NGDA*16] NISHIDA G., GARCIA-DORADO I., ALIAGA D. G., BENES B., BOUSSEAU A.: Interactive sketching of urban procedural models. *ACM Trans. on Graphics (TOG) 35*, 4 (2016), 130. 11

[OCDD15] ORTIZ-CAYON R., DJELOUAH A., DRETTAKIS G.: A Bayesian Approach for Selective Image-Based Rendering using Superpixels. In *Int. Conference on 3D Vision - 3DV* (Oct. 2015). 1, 3, 9, 10

[OCDM*16] ORTIZ-CAYON R., DJELOUAH A., MASSA F., AUBRY M., DRETTAKIS G.: Automatic 3d car model alignment for mixed image-based rendering. In *Int. Conference on 3D Vision (3DV)* (2016). 3, 11

[OSC14] OSWALD M. R., STÜHMER J., CREMERS D.: Generalized connectivity constraints for spatio-temporal 3d reconstruction. In *ECCV* (2014), Springer, pp. 32–46. 3

[PBCL09] PARK M., BROCKLEHURST K., COLLINS R. T., LIU Y.: Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Trans. PAMI 31*, 10 (2009), 1804–1816. 2

[PBCL10] PARK M., BROCKLEHURST K., COLLINS R. T., LIU Y.: Image de-fencing revisited. In *ACCV* (2010), Springer, pp. 422–434. 2

[PZ17] PENNER E., ZHANG L.: Soft 3d reconstruction for view synthesis. *ACM Trans. on Graphics (TOG) 36*, 6 (2017), 235. 1, 3, 8, 9, 10

[Rea16] REALITYCAPTURE C.: Realitycapture, 2016. 3, 9

[SCD*06] SEITZ S. M., CURLESS B., DIEBEL J., SCHARSTEIN D., SZELISKI R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR* (2006), vol. 1, pp. 519–528. 5

[SGHS98] SHADE J., GORTLER S., HE L.-W., SZELISKI R.: Layered depth images. In *SIGGRAPH'98* (1998), ACM, pp. 231–242. 3

[SKG*12] SINHA S. N., KOPF J., GOESELE M., SCHARSTEIN D., SZELISKI R.: Image-based rendering for scenes with reflections. *ACM Trans. Graph.* (2012). 1, 3

[TTVG01] TURINA A., TUYTELAARS T., VAN GOOL L.: Efficient grouping under perspective skew. In *CVPR* (2001), vol. 1, IEEE, pp. I–247. 2

[UB13] UMMENHOFER B., BROX T.: Point-based 3d reconstruction of thin objects. In *ICCV* (2013), pp. 969–976. 3

[WFP10] WU C., FRAHM J.-M., POLLEFEYS M.: Detecting large repetitive structures with salient boundaries. In *ECCV* (2010), Springer, pp. 142–155. 2

[WFZ02] WEXLER Y., FITZGIBBON A., ZISSERMAN A.: Bayesian estimation of layers from multiple images. In *ECCV* (2002), Springer, pp. 487–501. 2, 4, 5

[XRLF15] XUE T., RUBINSTEIN M., LIU C., FREEMAN W. T.: A computational approach for obstruction-free photography. *ACM Trans. on Graphics (TOG) 34*, 4 (2015), 79. 2, 9, 11

[YMK10] YAMASHITA A., MATSUI A., KANEKO T.: Fence removal from multi-focus images. In *ICPR* (2010), IEEE, pp. 4532–4535. 2

[YWT16] YI R., WANG J., TAN P.: Automatic fence segmentation in videos of dynamic scenes. In *CVPR* (2016), pp. 705–713. 2, 9, 11